Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently Amended) A data processing method for creating an executable file by combining a plurality of run units, the <u>data processing</u> method comprising:

<u>compiling, by an Assembler compiler, a plurality of Assembler modules to produce the plurality of run units, wherein each of the plurality of run units comprises a plurality of records corresponding to a plurality of Assembler instructions of an Assembler module, wherein each record of the plurality of records in a run unit of the plurality of run units begins with a key value identifying a specific type of Assembler instruction, wherein at least two types of Assembler instructions include a character string constant, and wherein at least two of the plurality of Assembler modules include at least one DL Assembler instruction;</u>

identifying<u>, by a linkage editor,</u> a first data entity <u>in a first run unit of the plurality of run units using a first DL Assembler instruction,</u> and a second data entity <u>in a second run unit of the plurality of run units using a second DL Assembler instruction,</u> ~~wherein both the first data entity and the second data entity are identified using an Assembler instruction, and~~ wherein the <u>first DL</u> Assembler instruction <u>and the second DL Assembler instruction</u> ~~identifies~~ <u>identify</u> character strings which are required to appear only once in the executable file<u>, further comprising:</u>

<u>setting the first data entity to a first value included in the first DL Assembler instruction, and setting the second data entity to a second value included in the second DL Assembler instruction;</u>

responsive to ~~a determination that a first~~ <u>the linkage editor receiving the second</u> run unit of the plurality of run units, ~~to be added to the executable file comprises the first~~ <u>comprising the second</u> data entity<u>, to be added the executable file</u> ~~set to a first value indicating that the first data entity is required to appear only once in the executable file,~~ determining whether the first data entity matches the second data entity ~~set to a second value and included in a second run unit~~, wherein ~~the second run unit comprises a run unit that~~ <u>the first run unit of the plurality of run units, comprising the first data entity,</u> was previously added to the executable file;

responsive to a determination that the first data entity matches the second data entity, adding the [[first]] second run unit of the plurality of run units to the executable file without the [[first]] second data entity; and

responsive to a determination that the first data entity does not match the second data entity, adding the [[first]] second run unit of the plurality of run units to the executable file with the [[first]] second data entity.

2.   (Currently Amended) [[A]] The data processing method of claim 1, wherein the first data entity matches the second data entity if the first value and second value are identical.

3.   (Currently Amended) [[A]] The data processing method of claim 1, wherein the first data entity matches the second data entity if the second value partially matches the first value.

4.   (Currently Amended) [[A]] The data processing method of claim 3, further comprising:

identifying a third data entity using [[an]] a third DL Assembler instruction that identifies character strings which are required to appear only once in the executable file, wherein the third data entity is set to a third value included in the third DL Assembler instruction;

determining whether the first data entity partially matches the third data entity included in a third run unit of the plurality of run units to be added to the executable file, wherein the third data entity is set to a third value indicating that the third data entity is required to appear only once in the executable file, and wherein the first data entity partially matches the third data entity if the third value contains the first value;

responsive to a determination that the first data entity partially matches the third data entity, removing the first data entity from the executable file; and

adding the third run unit of the plurality of run units comprising the third data entity to the executable file.

5.   (Currently Amended) [[A]] The data processing method of claim 1, wherein identifying a first data entity in a first run unit of the plurality of run units using a first DL Assembler instruction the determination that the first run unit to be added to the executable file comprises

~~the first data entity set to a first value indicating that the first data entity is required to appear~~ ~~only once in the executable file,~~ <u>further</u> comprises:

locating a <u>plurality of data entity records from the</u> plurality of <u>records</u> ~~data entities~~ in the first run unit <u>using each key value of the plurality of records in the first run unit to identify a DL type of Assembler instruction by which each of a plurality of data entities is marked, wherein each record of the plurality of records in the first run unit with a key value that identifies the DL type of Assembler instruction is identified as a data entity record in the plurality of data entity records, and wherein each record of the plurality of records in the first run unit with a key value that does not identify the DL type of Assembler instruction is not identified as a data entity record in the plurality of data entity records</u>; and

creating the first data entity [[from]] <u>by combining</u> the plurality of data entities.

6-20.   (Canceled)

21.   (Currently Amended) [[A]] <u>The data processing</u> method of claim 1<u>,</u> wherein <u>identifying a first data entity in a first run unit of the plurality of run units using a first</u> ~~the Assembler~~ ~~instruction is a~~ DL Assembler instruction <u>further comprises locating a first record for the first DL Assembler instruction in the first run unit of the plurality of run units using each key value of the plurality of records in the first run unit to identify a DL type of Assembler instruction, wherein the first record contains the first value of the first data entity; and</u>

<u>wherein identifying a second data entity in a second run unit of the plurality of run units using a second DL Assembler instruction further comprises locating a second record for the second DL Assembler instruction in the second run unit of the plurality of run units using each key value of the plurality of records in the second run unit to identify a DL type of Assembler instruction, wherein the second record contains the second value of the second data entity</u>.

22.   (Currently Amended) [[A]] <u>The data processing</u> method of claim 21, wherein the DL ~~Assembler instruction is a~~ type of Assembler instruction [[that]] denotes a non-executable data entity which needs only be included once in the executable file.

23.   (New) A data processing apparatus for creating an executable file by combining a plurality of run units, the data processing apparatus comprising:

a memory for storing a plurality of Assembler modules;

an Assembler compiler for compiling the plurality of Assembler modules to produce the plurality of run units, wherein each of the plurality of run units comprises a plurality of records corresponding to a plurality of Assembler instructions of an Assembler module, wherein each record of the plurality of records in a run unit of the plurality of run units begins with a key value identifying a specific type of Assembler instruction, wherein at least two types of Assembler instructions include a character string constant, and wherein at least two of the plurality of Assembler modules include at least one DL Assembler instruction;

a linkage editor for identifying a first data entity in a first run unit of the plurality of run units using a first DL Assembler instruction, and a second data entity in a second run unit of the plurality of run units using a second DL Assembler instruction, wherein the first DL Assembler instruction and the second DL Assembler instruction identify character strings which are required to appear only once in the executable file, further comprising:

   the linkage editor for setting the first data entity to a first value included in the
   first DL Assembler instruction, and setting the second data entity to a second value
   included in the second DL Assembler instruction;

the linkage editor for determining whether the first data entity matches the second data entity, wherein the first run unit of the plurality of run units, comprising the first data entity, was previously added to the executable file responsive to the linkage editor receiving the second run unit of the plurality of run units, comprising the second data entity, to be added the executable file;

the linkage editor for adding the second run unit of the plurality of run units to the executable file without the second data entity responsive to a determination that the first data entity matches the second data entity; and

the linkage editor for adding the second run unit of the plurality of run units to the executable file with the second data entity responsive to a determination that the first data entity does not match the second data entity.

24. (New) The data processing apparatus of claim 23, wherein the first data entity matches the second data entity if the second value partially matches the first value.

25. (New) The data processing apparatus of claim 24, further comprising the linkage editor for:

identifying a third data entity using a third DL Assembler instruction that identifies character strings which are required to appear only once in the executable file, wherein the third data entity is set to a third value included in the third DL Assembler instruction;

determining whether the first data entity partially matches the third data entity included in a third run unit of the plurality of run units to be added to the executable file, wherein the first data entity partially matches the third data entity if the third value contains the first value;

responsive to a determination that the first data entity partially matches the third data entity, removing the first data entity from the executable file; and

adding the third run unit of the plurality of run units comprising the third data entity to the executable file.

26. (New) The data processing apparatus of claim 23, wherein identifying a first data entity in a first run unit of the plurality of run units using a first DL Assembler instruction, further comprises the linkage editor for:

locating a plurality of data entity records from the plurality of records in the first run unit using each key value of the plurality of records in the first run unit to identify a DL type of Assembler instruction by which each of a plurality of data entities is marked, wherein each record of the plurality of records in the first run unit with a key value that identifies the DL type of Assembler instruction is identified as a data entity record in the plurality of data entity records, and wherein each record of the plurality of records in the first run unit with a key value that does not identify the DL type of Assembler instruction is not identified as a data entity record in the plurality of data entity records; and

creating the first data entity by combining the plurality of data entities.

27. (New) The data processing apparatus of claim 23, wherein identifying a first data entity in a first run unit of the plurality of run units using a first DL Assembler instruction further

comprises locating a first record for the first DL Assembler instruction in the first run unit of the plurality of run units using each key value of the plurality of records in the first run unit to identify a DL type of Assembler instruction, wherein the first record contains the first value of the first data entity; and

wherein identifying a second data entity in a second run unit of the plurality of run units using a second DL Assembler instruction further comprises locating a second record for the second DL Assembler instruction in the second run unit of the plurality of run units using each key value of the plurality of records in the second run unit to identify a DL type of Assembler instruction, wherein the second record contains the second value of the second data entity.

28.   (New) The data processing apparatus of claim 27, wherein the DL type of Assembler instruction denotes a non-executable data entity which needs only be included once in the executable file.

29.   (New) A computer program product for creating an executable file by combining a plurality of run units, the computer program product comprising:

a plurality of instructions stored in a memory, the plurality of instructions adapted to cause a processor of a data processing host to perform the steps of:

compiling, by an Assembler compiler, a plurality of Assembler modules to produce the plurality of run units, wherein each of the plurality of run units comprises a plurality of records corresponding to a plurality of Assembler instructions of an Assembler module, wherein each record of the plurality of records in a run unit of the plurality of run units begins with a key value identifying a specific type of Assembler instruction, wherein at least two types of Assembler instructions include a character string constant, and wherein at least two of the plurality of Assembler modules include at least one DL Assembler instruction;

identifying, by a linkage editor, a first data entity in a first run unit of the plurality of run units using a first DL Assembler instruction, and a second data entity in a second run unit of the plurality of run units using a second DL Assembler instruction, wherein the first DL Assembler instruction and the second DL Assembler instruction identify character strings which are required to appear only once in the executable file, further comprising:

setting the first data entity to a first value included in the first DL Assembler instruction, and setting the second data entity to a second value included in the second DL Assembler instruction;

responsive to the linkage editor receiving the second run unit of the plurality of run units, comprising the second data entity, to be added the executable file, determining whether the first data entity matches the second data entity the first run unit of the plurality of run units, comprising the first data entity, was previously added to the executable file;

responsive to a determination that the first data entity matches the second data entity, adding the second run unit of the plurality of run units to the executable file without the second data entity; and

responsive to a determination that the first data entity does not match the second data entity, adding the second run unit of the plurality of run units to the executable file with the second data entity.

30.   (Currently Amended) The computer program product of claim 29, wherein the first data entity matches the second data entity if the first value and second value are identical.

31.   (Currently Amended) The computer program product of claim 29, wherein the first data entity matches the second data entity if the second value partially matches the first value.

32.   (New) The computer program product of claim 31, further comprising the plurality of instructions adapted to cause a processor of a data processing host to perform the steps of:

identifying a third data entity using a third DL Assembler instruction that identifies character strings which are required to appear only once in the executable file, wherein the third data entity is set to a third value included in the third DL Assembler instruction;

determining whether the first data entity partially matches the third data entity included in a third run unit of the plurality of run units to be added to the executable file, wherein the first data entity partially matches the third data entity if the third value contains the first value;

responsive to a determination that the first data entity partially matches the third data entity, removing the first data entity from the executable file; and

adding the third run unit of the plurality of run units comprising the third data entity to the executable file.

33.   (New) The computer program product of claim 29, wherein identifying a first data entity in a first run unit of the plurality of run units using a first DL Assembler instruction, further comprises the plurality of instructions adapted to cause the processor of the data processing host to perform steps of:

locating a plurality of data entity records from the plurality of records in the first run unit using each key value of the plurality of records in the first run unit to identify a DL type of Assembler instruction by which each of a plurality of data entities is marked, wherein each record of the plurality of records in the first run unit with a key value that identifies the DL type of Assembler instruction is identified as a data entity record in the plurality of data entity records, and wherein each record of the plurality of records in the first run unit with a key value that does not identify the DL type of Assembler instruction is not identified as a data entity record in the plurality of data entity records; and

creating the first data entity by combining the plurality of data entities.

34.   (New) The computer program product of claim 29, wherein identifying a first data entity in a first run unit of the plurality of run units using a first DL Assembler instruction further comprises locating a first record for the first DL Assembler instruction in the first run unit of the plurality of run units using each key value of the plurality of records in the first run unit to identify a DL type of Assembler instruction, wherein the first record contains the first value of the first data entity; and

wherein identifying a second data entity in a second run unit of the plurality of run units using a second DL Assembler instruction further comprises locating a second record for the second DL Assembler instruction in the second run unit of the plurality of run units using each key value of the plurality of records in the second run unit to identify a DL type of Assembler instruction, wherein the second record contains the second value of the second data entity.

35.   (New) The computer program product of claim 34, wherein the DL type of Assembler instruction denotes a non-executable data entity which needs only be included once in the executable file.